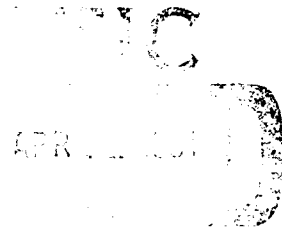


AD-A234 504

CAR-TR-545
CS-TR-2639N00014-91-J-1222
March 1991HIGH ORDER DIFFERENTIATION FILTERS
THAT WORK

Isaac Weiss

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3411

Abstract

Reliable derivatives of digital images have always been hard to obtain, especially (but not only) at high orders. We present new filters that give more accurate derivatives than the traditional Gaussian ones. We show that the traditional filters give incorrect derivatives even for an analytic, noiseless, infinite image, because they smooth the image too much. For a finite interval, the effects of truncating the filter become intolerable for high derivatives. We derive filters that allow a higher amount of noise suppression with less compromise of accuracy than the Gaussian. The filters are easy to compute at arbitrary size. In addition, a general analytic (non-filter) solution is derived for the regularization problem on a finite interval.

Keywords: smoothing, differentiation, regularization, filters, fitting.

| | |
|----------|---|
| Author | ✓ |
| Title | |
| Subject | |
| Indexing | |
| By | |
| Date | |
| File | |
| Notes | |
| Comments | |

A-1

The author is supported in part by ONR Grant N00014-91-J 1222.

91 4 10 132

1. Introduction

Finding derivatives in an image has always been a desirable goal in computer vision because it is often the local changes in the scene that characterize the shape, for instance its curvature, which is seen as a change in shading or in other image characteristics. Curvature itself involves second derivatives, and finding maxima of curvature requires third ones. In some applications, for instance in finding affine and projective invariants of shapes [Weiss, 1988, 1991], even higher derivatives are required. Thus, having reliable derivatives makes it possible to apply the vast body of knowledge that exists about differential mathematical methods.

The usual methods for finding derivatives have been very unreliable, with the problems growing unacceptably worse with higher derivatives. This has severely limited the usefulness of most methods in vision that rely on derivatives for their application. A tendency has developed in the field to avoid the problem altogether and try to find alternative methods, but the need for derivatives has not simply gone away. Particularly for smooth shapes without sharp edges, there is no good substitute for differential methods. We are not dealing here with discontinuities. Rather, we are interested in higher derivatives of smooth shapes. However, our method can pinpoint the location of zero crossings in a smoothed image with accuracy, whereas a simple Gaussian smoothing tends to dislocate them. This can be important in recognizing shapes with scale space methods.

In this paper we analyze the sources of errors in common methods for differentiation and show how to correct them. We start with some basic requirements that we would like differentiation operators to satisfy and show how to build them while avoiding the usual problems. Two basic, and conflicting, requirements are involved: a) Accuracy: At least for smooth, noiseless image, one would like to obtain the correct derivatives, at least at low orders. b) Smoothing, to alleviate the effect of noise and discretization. Generally, smoothing reduces the accuracy even in the analytic case as the more rapid changes in the function are smoothed out. The goal in designing a derivative filter is then to strike the correct balance between accuracy and smoothing. In this paper we develop filters that

yield correct low order derivatives (up to a desired order), while the higher derivatives are reduced in a controlled way by the smoothing parameter. This combats noise because noise changes considerably from pixel to pixel and thus contributes mainly to the high order variations, while it tends to average out in the larger scale, so it does not add much to the slower variations in the image. The method is based on a modification of the regularization method combined with a spline approximation. The results are rather simple closed-form expressions for the filters.

Most existing methods are based on smoothing the shape with a Gaussian filter and then taking the derivatives of the smoothed image. This is equivalent to filtering with a derivative of the Gaussian. We show that this method gives the wrong results even for the low order derivative of a simple analytic polynomial. The smoothing effect is too strong relative to the accuracy needed for most applications. In addition, truncating the Gaussian at finite boundaries usually adds avoidable errors, which are quite benign for smoothing but can be particularly damaging for high derivatives and lead to meaningless results even in noiseless images. This is because truncation is equivalent to introducing a sharp discontinuity in the data which makes the higher derivatives meaningless. It turns out that this problem can be solved simply by replacing the derivative of the Gaussian (or other smoothing function) by the central difference of the same order. This amounts to a spline approximation of the same order. A more complicated approach based on a general analytic solution of the regularization equation is also presented.

Other methods that have been used can be divided into several classes: 1) Methods that produce small but easy to calculate filters (Hueckel, [1973], Dierckx, [1977], Hummel, [1979], Haralick, [1984], Besl, [1988]). They are usually quite *ad hoc* and the results are not very reliable. 2) Methods based on global regularization, that assign an unknown variable to each pixel and solve a large system of equations for the whole set of pixels in the image (Grimson, [1981], Horn, [1983], Poggio, [1987], Kass, Witkin, and Terzopoulos, [1987], Blake and Zisserman, [1987], Poggio *et al.* [1988]. These methods, being based on sound principles, are rather robust but they are computationally intensive, and the

accuracy of their derivatives has not been investigated. 3) Method occupying the middle ground between the two extremes. Meer and Weiss [1989] used polynomial-based filters that smooth the image over a given size window which is moved over the image. The least squares distance is found between the image and a finite series of orthogonal polynomials; the degree of smoothing is determined by the highest order of polynomial used. Similar filters were discussed by [Hashimoto and Sklansky, 1987]. While these filters work well in many cases, the high derivative filters tend to concentrate in the center of the window, reducing the smoothing. Weiss [1989] introduced a regularization based method, again for use in moving finite windows. While the amount of calculation needed is orders of magnitude less than in traditional regularization, it is still more complicated than one would like. The present method is based on a continuous approximation so it is much easier to derive the filters. One can choose the window size and the smoothing parameter and then easily calculate the differentiation filter.

In the next section a general criterion for differentiation filters will be developed. In subsequent sections we will then derive smoothing filters for a finite window, deal with differentiation, and show experimental results.

2. Smoothing Versus Accuracy

In this section we analyze the balance between smoothing and accuracy for a general filter, and derive a general "accuracy criterion" that a good filter should satisfy.

As an illustration we first show that the Gaussian gives the wrong result even for the simplest functions. Smoothing over x^2 gives

$$g(x, \sigma) \odot x^2 = x^2 + \sigma^2$$

where $g(x, \sigma)$ is the Gaussian, and \odot denotes a convolution. Smoothing x^3 gives

$$g(x, \sigma) \odot x^3 = x^3 + 3\sigma^2 x$$

We can see that an error is introduced that increases as we increase the smoothing, i.e. higher σ . Similar results are obtained for higher powers or for taking derivatives. For

effective smoothing σ should not be too small so this error can be substantial. This is a *systematic*, “*oversmoothing error*”, one in the larger scale, as opposed the small scale random noise that we want to smooth. As noted before, we do not expect a smoothing operator to give accurate results but we do want to improve the balance between accuracy and smoothing so that at least the leading terms in a Taylor expansion will not produce errors.

We will now discuss this noise versus signal problem in terms of the ratio between the smoothing parameter σ , and a “natural” scale of the shape, s_0 . For a smooth shape $f(x)$, we estimate this s_0 as the scale over which the relative change in the shape, $\Delta f/f$, is of order of magnitude of 1. We then rescale the x axis so that $x_1 = x/s_0$, and rewrite the shape as $\bar{f}(x_1)$. For example, if $f(x) = \sin(x/s_0)$ then $\bar{f}(x_1) = \sin(x_1)$. This makes the derivatives of \bar{f} with respect to x_1 of the order of magnitude of 1. In this way we “normalize” the signal in the x direction and deal with its scale separately.

Intuitively speaking, when smoothing the image of this shape, we would like to maintain the accuracy of changes on scale s_0 and longer scales, but smooth the smaller features in the image. This is because the shape is assumed to be smooth on the shorter scales and any changes there is probably noise. In the following we will quantify this concept.

2.1 Accuracy Criterion. A natural tool in dealing with derivative in some neighborhood of a smooth shape is the Taylor expansion

$$f(x) = \bar{f}\left(\frac{x}{s_0}\right) = \sum_{\nu} \frac{\bar{f}^{(\nu)}}{\nu!} \left(\frac{x}{s_0}\right)^{\nu}$$

We will look at the result of Gaussian filtering at $x = 0$. It is easy to show that the error introduced by this smoothing is

$$g \otimes f - f \approx \bar{f}''\left(\frac{\sigma}{s_0}\right)^2 + \frac{1}{8}\bar{f}^{(4)}\left(\frac{\sigma}{s_0}\right)^4 + \dots$$

with the derivatives $\bar{f}^{(\nu)} = \frac{d^{\nu} \bar{f}}{d(x/s_0)^{\nu}} \approx O(1)$. If we want accurate results, we have to keep this error small. Looking at the leading term, we see that the accuracy is proportional to

σ^2/s_0^2 , so we have to keep σ small, i.e. $\sigma \ll s_0$. Unfortunately, this limits the ability of the filter to smooth out the noise.

For derivatives we similarly have (with $g^{(k)} \otimes f = g \otimes f^{(k)}$)

$$g^{(k)} \otimes f - f^{(k)} \approx \frac{1}{s_0^k} \left[\bar{f}^{(k+2)} \left(\frac{\sigma}{s_0} \right)^2 + \frac{1}{8} \bar{f}^{(k+4)} \left(\frac{\sigma}{s_0} \right)^4 + \dots \right]$$

Again the error resulting from the leading term is proportional to σ^2/s_0^2 . (For the relative error we can divide this by the true derivative, $f^{(k)} = \bar{f}^{(k)}/s_0^k$.)

A way to improve the situation is to eliminate the leading terms in the expansions of the errors above. If the first term is eliminated, for instance, then the error will be reduced to $\approx \frac{1}{8} \left(\frac{\sigma}{s_0} \right)^4$. This way we obtain a much better accuracy for the same smoothing parameter σ as before, or alternatively, we can increase the smoothing without compromising accuracy.

To generalize this example, we want a filter F_l that will preserve the powers x^n :

$$F_l \otimes x^n = x^n, \quad n = 0 \dots l$$

(For the Gaussian $l = 1$.) These conditions can be expressed more simply in terms of the filter's "normalized moments" m_n :

$$m_n = \int \left(\frac{x}{\sigma} \right)^n F_l(x) dx$$

where σ is now a measure of the filter size, e.g. the variance. Using the binomial expansion we have

$$\begin{aligned} F_l \otimes x^n &= \int F_l(x - \xi) (\xi - x + x)^n d\xi \\ &= \int F_l(\xi - x) [(\xi - x)^n + n(\xi - x)^{n-1}x + \dots + x^n] d(\xi - x) \\ &= m_n \sigma^n + n m_{n-1} \sigma^{n-1} x + \dots + m_0 x^n \end{aligned}$$

We see that the preservation of powers x^n by the convolution is equivalent to the conditions on the moments:

$$m_0 = 1; \quad m_n = 0, \quad n = 1 \dots l$$

The error in the smoothing of $f^{(k)}$ at $x = 0$ can now be written as

$$F_l \otimes f^{(k)} - f^{(k)} \approx \frac{m_{l+1}}{(l+1)!} \bar{f}^{(l+k+1)} \left(\frac{\sigma}{s_0} \right)^{l+1} + \dots$$

Thus, increasing the order l of the filter eliminate the powers $(\frac{\sigma}{s_0})^n$ up to $n = l$ and improve the accuracy to the above tolerance.

The error expression above gives us an “accuracy criterion” for choosing appropriate parameters. We need its leading term to be small, and since $\bar{f}^{(\nu)} \approx O(1)$ we obtain the condition

$$\frac{m_{l+1}}{(l+1)!} \left(\frac{\sigma}{s_0} \right)^{l+1} \ll 1 \quad (1)$$

This criterion can be used to estimate the parameters in several ways. Given the meaningful scale of change of the signal, s_0 , and the smoothing parameter σ , we can calculate the order l of the filter that will lower the oversmoothing error to some acceptable level. σ can be determined from the estimated noise, as discussed below. Conversely, for a given order l , we can calculate the largest smoothing parameter σ that will still maintain a desired accuracy. Generally speaking, at low orders we need $\sigma < s_0$, but at high orders we can afford to have the smoothing σ bigger than s_0 because the factor $m_{l+1}/(l+1)!$ in (1) is small.

One way of obtaining such x^n -preserving filters is by multiplying a Gaussian with appropriate polynomials. The filter of order l , eliminating errors up order l , is then (on an infinite interval)

$$F_l = \sum_{i=0}^l a_i P_i(x) g(x)$$

where $P_i(x)$ are Hermite polynomials which are orthogonal with respect to the Gaussian weight function. The coefficients a_i are chosen so that the first l powers x^n are preserved by the filter. Discrete versions of this method on a finite window are described in detail in [Meer and Weiss, 1989]. A different way to achieve this goal is described here, which has advantages with respect to handling noise.

2.2 Noise Suppression. Maintaining accuracy is only one side of the balance we are interested in. The other side is suppressing noise. We call noise any random variation including “bumps” in the shape, discretization noise, etc. The noise is not polynomial and can better be described by a Fourier representation, $N(\omega)$. The derivative of this noise contributes the the error in finding derivative of the shape. After smoothing the shape by the filter F_l the noise derivative in the Fourier domain is

$$N^{(k)}(\omega) = \omega^k \bar{F}_l(\sigma\omega) N(\omega)$$

with \bar{F} being the Fourier transform of F . We can write this error in terms of scale of change s :

$$s = \frac{1}{\omega}$$

For the signal to noise ratio of the derivative we can divide this error by the derivative itself, $f^{(k)} = \bar{f}^{(k)}/s_0^k$, to obtain

$$\frac{N^{(k)}}{f^{(k)}} \approx N\left(\frac{1}{s}\right) \left(\frac{s_0}{s}\right)^k \bar{F}_l\left(\frac{\sigma}{s}\right)$$

Since we assume that the noise N vanishes above the scale $s > s_0$, (i.e. any change above this scale is part of the signal), we want the filter to do no smoothing there. However, for $s < s_0$ we see that this error is proportional \bar{F}/s^k and smoothing becomes critical. Without smoothing ($\bar{F} = 1$), it will tend to infinity as the size of the bumps s tends to zero. (This can also be seen by direct differentiation of N). The effectiveness of the smoothing is thus determined by the factor $\bar{F}_l(\sigma/s)$, so we need a filter that decays rapidly in the Fourier domain as a function of σ/s . This implies a requirement for large σ . Generally speaking, a filter will smooth noise bumps whose size and height are small compared with σ . We thus want σ to be large enough to make the noise to signal ratio above small, but not large enough to violate the accuracy criterion (1). This criterion shows that we can increase the smoothing parameter with little compromise of accuracy by using higher order filters.

For the Gaussian, this smoothing factor is $e^{-\sigma^2/s^2}$, so it will suppress any derivative of the noise with scale smaller than σ . However, it also suppresses the changes on the larger

scales $s > s_0$ which are not noise, and this is another expression of the oversmoothing problem discussed above. (In terms of the previous analysis, the accuracy criterion will be met only with a very small σ). We need a filter that will not affect the changes on scale $s > s_0$ but will suppress the changes on scales $s < s_0$. In this paper we will derive filters, based on regularization theory, which meet these requirements.

The regularization filter for the *infinite* line is a Butterworth-like filter: (Sec. 3)

$$\bar{F}_l = \frac{1}{1 + (\sigma/s)^{l+1}}$$

(with l being the order of the filter). This is in fact a good low pass filter (Fig. 3), preserving the slow changes and suppressing the faster changes in derivatives up to order l . Here we will modify this filter for a finite interval. This is important for finding derivatives, which are of a local nature. In the case of the Hermite and other polynomial filters, \bar{F}_l start to decay at a certain point but then they oscillate for a long interval, so they can have more difficulty dealing with smoothing noisy derivatives.

In summary, we have seen some sources of errors in differentiation, and found the requirements that a good filter has to meet. Using both Taylor expansion analysis and Fourier analysis we have concluded: (i) the Gaussian filter oversmooths the signal, (ii) other filters can be derived which do not have this problem, and (iii) an appropriate choice of the parameters of these filters is the key to obtaining a good signal to noise ratio in the derivatives. Some general guidelines were found (the accuracy criterion (1)) for this choice, with the details depending on the specific filter.

The above conclusions are valid for both finite and infinite intervals. However, the finite case can add other serious errors that we deal with later.

3. Regularized Differentiation on the Infinite Line

As an introduction to our method we treat here an infinite interval. The next section treats the finite interval case, more suitable for derivatives. In the previous section we described the need to improve the balance between smoothing and accuracy provided by the Gaussian. A method that allows us to adjust this balance to fit our needs is the regularization method. It basically consists of minimizing a cost function that contains both an accuracy term and a smoothing term. The cost functional can be written as

$$\int ((f(x) - p(x))^2 + \lambda f''(x)^2) dx$$

where $p(x)$ is the image data and $f(x)$ is the smoothed shape. The first term is a measure of the square distance of the smoothed shape from the data, while the second represents the variation, or unsmoothness, of the shape. The parameter λ determines the balance between smoothing and accuracy. In the continuous case, it is possible to minimize this function by using the Euler-Lagrange equation of the calculus of variations, which leads to the following differential equation for the shape f

$$f(x) + \lambda f''''(x) = p(x) \quad (2)$$

This equation can be solved instantly for the simple data functions $p = x^n$, $n < 4$. In this case $f = p$ (because the fourth derivative vanishes). That is, the smoothed shape is equal to the data image in this case, unlike the results of Gaussian smoothing that we saw earlier. Thus, with a similar degree of smoothing, the regularization method produces more accurate values of the first powers x^n than the Gaussian.

The next step is to produce a general purpose filter based on the above method, so that we will not have to solve a fourth-order differential equation for every image. To do this we use the Green function method of differential equations. We solve for a "needle"-like image, i.e. a delta function $\delta(x - \xi)$, where ξ is any given point in the image. The solution is our Green function. Because of the linearity of the problem, a general solution can be built from such basis solutions.

In a physical analogy, our cost function represents a flexible "snake" with the smoothness parameter corresponding to the stiffness of the material. The snake can be made up of a combination of "basis" snakes. Each basis snake is a smooth version of one data point, or a "needle" image. Since the the problem is linear, a combination of such snakes will give the appropriate smooth shape.

More formally, denoting the Green function by G , we want it to solve

$$G(x - \xi) + \lambda G''''(x - \xi) = \delta(x - \xi) \quad (3)$$

The solution for general data is now

$$f(x) = \int G(x - \xi)p(\xi)d\xi \quad (4)$$

The above solution is in fact a convolution of the image data p with a filter, the Green function G .

The simplest way to find this Green function is via a Fourier transform. Transforming both sides of eq. (3) we have, with $G(\omega)$ being the transform of $G(x)$

$$G(\omega) + \lambda \omega^4 G(\omega) = 1$$

from which $G(\omega)$ is

$$G(\omega) = \frac{1}{1 + \lambda \omega^4}$$

Transforming back to the coordinate domain we have to perform the integral

$$G(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{e^{i\omega x}}{1 + \lambda \omega^4} d\omega$$

The derivatives $G^{(n)}(x)$ can be found directly or from

$$G^{(n)}(x) = \frac{\partial^n G(x)}{\partial x^n} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{i^n \omega^n e^{i\omega x}}{1 + \lambda \omega^4} d\omega$$

The above integrations can be performed using the residue method of analytic functions. As expected from the theory of differential equations, the result is a function with one discontinuity, at $x = 0$:

$$G^{(n)}(x) = \begin{cases} G_+^{(n)}(x) & \text{for } x \geq 0 \\ (-1)^n G_+^{(n)}(-x) & \text{for } x < 0 \end{cases}$$

For $n = 0$, the smoothing filter, we have $G(x) = G_+(|x|)$ with

$$G_+ \equiv G_+^{(0)}(x) = \frac{1}{2\sigma} e^{-x/\sigma} (\sin(x/\sigma) + \cos(x/\sigma))$$

where

$$\sigma \equiv (\lambda/4)^{1/4}$$

We will also need the integral of G :

$$G_+^{(-1)}(x) = \int_0^x G(x) = \frac{1}{2} [1 - e^{-x/\sigma} \cos(x/\sigma)]$$

The smoothing filter G is plotted in Fig. 1 along with a Gaussian with the same σ . The first derivative filter G' is plotted in Fig. 2 with the derivative of the Gaussian. Figs. 3,4 compare the Fourier transforms of the smoothing and differentiation filters, respectively. We can see from the transform of the smoothing filter G that it acts like a low-pass filter, with the low frequency response being almost flat, meaning that the slower, larger-scale variations in the data are preserved much better than they are by the Gaussian. Smoothing begins only above a certain frequency, depending on the smoothing parameter, and increases for higher frequencies. This is in line with the earlier observation about the preservation of the low order Taylor terms and with our initial requirements on a differentiation operator.

It can be verified that the powers x^n , $n < 4$ are indeed eigenvalues of the convolution with this filter:

$$G \otimes x^n = \int_{-\infty}^{\infty} G(x - \xi) \xi^n d\xi = x^n, \quad n = 0, 1, 2, 3 \quad (5)$$

in line with our earlier observation about the preservation of powers x^n .

The same method can be carried to higher orders, so that eq. (5) will hold for higher powers n . As discussed in Section 2, a method that will preserve higher powers of x will allow using a stronger smoothing factor σ . To obtain that, we replace f'' in the cost function by $f^{(k)}$. This results in the following G_k :

$$G_k(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{e^{i\omega x}}{1 + \lambda\omega^{l+1}} d\omega$$

where $l = 2k - 1$ is the order of the filter, which will preserve powers up to x^l .

The filter with $k = 3$ (or $l = 5$) is, with $\sigma = \lambda^{\frac{1}{2k}}$,

$$G_{3,+} = \frac{1}{3\sigma} [e^{-x/\sigma} (\sqrt{3} \sin(\sqrt{3}x/\sigma) + \cos(\sqrt{3}x/\sigma)) + e^{-2x/\sigma}]$$

and for $k = 4$, (or $l = 7$) with

$$c_1 = \sqrt{(2 + \sqrt{2})}/2, \quad c_2 = \sqrt{(2 - \sqrt{2})}/2$$

we have

$$G_{4,+} = \frac{1}{4\sigma} [e^{-c_2 x/\sigma} (c_2 \cos(c_1 x/\sigma) + c_1 \sin(c_1 x/\sigma)) + e^{-c_1 x/\sigma} (c_1 \cos(c_2 x/\sigma) + c_2 \sin(c_2 x/\sigma))]$$

The infinite $k = 2$ case was also derived by [Poggio *et al.*, 1985] without noting the connection with preservation of powers. Here we shall deal with high order derivatives of a smooth shape in a finite neighborhood, and this is developed next.

4. The Finite Window Case: Preserving Powers

In practice one wants to use a finite size window, i.e. convolve the image with a finite filter not only because of physical limitations but because one does not want to smooth over discontinuities, so the size of the filter has to be smaller than the size of the smooth parts of the shape. Derivatives, in particular, are local phenomena and call for a local filter.

One is tempted to simply truncate an infinite filter at some finite point, but this is inappropriate for two reasons: (i) truncation amounts to introducing a discontinuity in the

function with disastrous effects on higher derivatives. This will be dealt with later. (ii) the property of preservation of powers takes a different form on a finite interval and the Green function has to be modified. This will be done now.

To maintain the accuracy of the filter we have to maintain the property that the filter will preserve the lower powers x^n on a finite interval, and for that we have to deal in more detail with the Green function method. This property will translate into linear conditions on the parameters of the Green function.

A general solution of a linear differential equation like (3) has two kinds of solutions: a homogeneous solution, which solves the left hand side (homogeneous) part of it alone (with zero r.h.s.), and an inhomogeneous part, that makes the left hand side equal to the right hand side. For our 4-th order equation one can express this as

$$G(x, \xi) = G_{\infty}(x - \xi) + \sum_{i=1}^4 b_i(\xi) H_i(x - x_0) \quad (6)$$

G_{∞} is the solution in the infinite case found before and it provides the δ function at $x = \xi$ (i.e. our "needle" is at ξ),

$$(1 + \lambda \partial^4) G_{\infty}(x - \xi) = \delta(x - \xi)$$

This is our inhomogeneous solution. The homogeneous solutions $b_i H_i$ (with b_i being arbitrary coefficients and H_i some basis homogeneous solutions) only solve the l.h.s. The origin x_0 can be chosen arbitrarily, and without loss of generality it can be chosen as zero, but here we will need another choice. The general solution is thus not unique and is determined by boundary conditions.

The infinite case was special in the sense that the homogeneous solution need not be considered, because the boundary conditions require that the solution and its derivatives vanish at infinity and only a trivial (zero) homogeneous solution has this property. Thus, G_{∞} in that case is a unique solution of eq. (3) and $G_{\infty} \circledast p$ is the unique solution for general data $p(x)$. So, with the data $p(x) = x^n$ ($n < 4$), the unique solution is $f(x) = x^4$ and it is give by this convolution. In the finite boundary case, there are many possible solutions

and the one given by $G_\infty \otimes p$ is x^4 "contaminated" by the addition of some homogeneous solution.

Our goal is now to find the coefficients b_i in the homogeneous solution in (6) that will restore the desirable properties we had in the infinite case, namely the preservation of x^n , $G \otimes x^n = x^n$.

To simplify the treatment, we assume now that it is enough to find the Green function on a small window rather than for the whole image. This window can then be moved across the image to find the whole solution. The justification is that the data outside the window does not influence the smoothed shape inside very much. We form the window such that its center is located at the "needle" position ξ and it extends to a width w around this center. This can be done by setting $x_0 = \xi$ in eq. (6), and setting the window boundaries at $\xi \pm w$. Since the windows are all similar the coefficients b_i are now constants and eq. (6) becomes shift-invariant:

$$G(x - \xi) = G_\infty(x - \xi) + \sum_{i=1}^4 b_i H_i(x - \xi)$$

The general solution of (2) is now the linear combination of windows:

$$f(x) = \int_{x-w}^{x+w} G(x - \xi) p(\xi) d\xi$$

Since eq. (2) is a fourth order equation we have four independent boundary conditions that we can impose. In line with our previous conditions on the moments we can choose

$$\int_{-w}^w G(x) = 1 \tag{7a}$$

$$\int_{-w}^w G(x) x^n = 0, \quad n = 1, 2, 3 \tag{7b}$$

It is easy to show that, with the shift invariance approximation above, these conditions lead to the the preservation of powers on a finite window, in analogy with the infinite case.

We will only show it for $n = 3$. We have (dropping the subscript from ξ):

$$\begin{aligned} G \otimes x^3 &= \int_{x-w}^{x+w} G(x-\xi) \xi^3 d\xi = \int_{x-w}^{x+w} G(x-\xi) (\xi - x + x)^3 d\xi \\ &= \int_{-w}^w G(\xi - x) [(\xi - x)^3 + 3(\xi - x)^2 x + 3(\xi - x)x^2 + x^3] d(\xi - x) \\ &= x^3 \end{aligned}$$

The last equality follows from the results for $n < 3$.

In practice, it appears that one can squeeze one more condition out of the system. Instead of normalizing the filter (condition (7a)) by adding the homogeneous solutions, the normalization can then be done directly, with a multiplying constant $1/N$. It amounts to finding the smoothed version of an image which is multiplied by the constant N , rather than the image itself. In fact, this kind of normalization can be done for other filters, such as a truncated Gaussian, as well. Eq. (7a) can now be replaced by some other condition. We can demand, for example, that the filter vanish at the ends of the window. The filter is of course zero outside the window, so this will provide continuity. Another possibility (not tried) is demanding the conservation of the x^4 moment, which will immediately preserve the x^5 moment too, because of its antisymmetry.

The boundary condition (7) give rise to linear equations for b_i which we will now solve. During this treatment we can replace $(x - \xi)/\sigma$ by x and the half-width of the window w is replaced by a normalized width a :

$$a = \frac{w}{\sigma}$$

We will obtain a simple linear system of two equations that need to be solved for a given a . The four independent homogeneous solutions can be written as

$$e^{\pm x} \cos(x), \quad e^{\pm x} \sin(x)$$

Condition (7b) with $n = 1, n = 3$ can be met immediately by choosing the two symmetric combinations

$$H_1(x) = \cosh(x) \cos(x), \quad H_2(x) = \sinh(x) \sin(x)$$

The Green function can now be written as

$$G(x) = (G_{\infty}(x) + b_1 H_1(x) + b_2 H_2(x))/N \quad (8)$$

with b_1, b_2, N to be found by the remaining conditions. From condition (7b) with $n = 2$ we find by integration by parts

$$\int_0^a x^2 G(x) = a^2 G^{(-1)}(a) - 2a G^{(-2)}(a) + 2G^{(-3)}(a) = 0$$

The negative superscripts mean integration rather than a derivative. Since it is easier to work with derivatives we will use

$$G^{(n)} = -4G^{(n-4)}$$

(which is the property that solves (3)) to obtain

$$G'(a) - aG''(a) + a^2 G'''(a)/2 = 0 \quad (9)$$

Interestingly, this is a Taylor expansion of $G'(x)$ around the point $x = a$, evaluated at $x = 0$, and it is equal to the exact value $G'(0, a) = 0$. It is not clear if this fact is of importance. Substituting (8) in the above we have

$$\begin{aligned} & G'_{\infty}(a) - aG''_{\infty}(a) + a^2 G'''_{\infty}(a)/2 \\ & + b1(H'_1(a) - aH''_1(a) + a^2 H'''_1(a)/2) \\ & + b2(H'_2(a) - aH''_2(a) + a^2 H'''_2(a)/2) = 0 \end{aligned} \quad (10a)$$

Together with the condition of vanishing at the end

$$G_{\infty}(a) + b_1 H_1(a) + b_2 H_2(a) = 0 \quad (10b)$$

we have a system of two linear equations for b_1, b_2 . All other quantities appearing in it are easily calculated so it can be solved. We collect the coefficients used in eq. (10a) for

reference:

$$G'_{\infty}(a) = -e^{-a} \sin(a)$$

$$G''_{\infty}(a) = e^{-a}(\sin(a) - \cos(a))$$

$$G'''_{\infty}(a) = 2e^{-a} \cos(a)$$

$$H'_1(a) = \sinh(a) \cos(a) - \cosh(a) \sin(a)$$

$$H'_2(a) = \sinh(a) \cos(a) + \cosh(a) \sin(a)$$

$$H''_1(a) = -2 \sinh(a) \sin(a)$$

$$H''_2(a) = 2 \cosh(a) \cos(a)$$

$$H'''_1(a) = -2 \sinh(a) \cos(a) - 2 \cosh(a) \sin(a)$$

$$H'''_2(a) = 2 \sinh(a) \cos(a) - 2 \cosh(a) \sin(a)$$

All that remains is to calculate the normalization constant N :

$$\begin{aligned} N &= 2(G_{\infty}^{(-1)}(a) + b_1 H_1^{(-1)}(a) + b_2 H_2^{(-1)}(a)) \\ &= 1 - e^{-a} \cos(a) + (b_1 + b_2) \cosh(a) \sin(a) + (b_1 - b_2) \sinh(a) \cos(a) \end{aligned}$$

In summary, after deciding on the parameter a , i.e. the ratio of the window size and the smoothing parameter σ , it is straightforward to calculate the parameters b_1, b_2, N and substitute in the filters $G^{(n)}(x)$. The filters used in our experiments are finally:

$$\begin{aligned} G_+(x) &= \frac{1}{\sigma N} [e^{-x/\sigma} (\cos(\frac{x}{\sigma}) + \sin(\frac{x}{\sigma}))/2 + b_1 (\cosh(\frac{x}{\sigma}) \cos(\frac{x}{\sigma}) + b_2 (\sinh(\frac{x}{\sigma}) \sin(\frac{x}{\sigma}))] \\ G_+^{(-1)}(x) &= \frac{1}{2N} [1 - e^{-x/\sigma} \cos(\frac{x}{\sigma}) + (b_1 + b_2) \cosh(\frac{x}{\sigma}) \sin(\frac{x}{\sigma}) + (b_1 - b_2) \sinh(\frac{x}{\sigma}) \cos(\frac{x}{\sigma})] \quad (11) \end{aligned}$$

The smoothing filter G is symmetric and vanishes outside the window, while its integral $G^{(-1)}$ is antisymmetric and is equal to $\pm 1/2$ on the upper/lower ends of the window.

Figs. 5,6 show the finite smoothing filters $G_+(x)$ for various σ s.

At higher orders, it may be easier to derive the finite filter by a polynomial expansion.

5. Rounding the Filter's Edges

In this section we show that for differentiation, the discontinuities of the filter, e.g. at the ends, can lead to significant errors even in the absence of noise, and show a way to overcome it. This treatment does not depend on the filter so it is valid for any filter. Truncating an infinite filter is often the cause of these discontinuities, but the finite filter developed before is also not continuous enough for use with derivatives.

The derivative G' can be used as a differentiation filter only when the smoothing filter G vanishes at the ends, because, by integration by parts,

$$\int_{-w}^w G(\xi)p'(x-\xi)d\xi = \int_{-w}^w G'(\xi)p(x-\xi)d\xi + G(w)(p(x+w) - p(x-w))$$

The last term is usually neglected when using a truncated Gaussian filter. Another way of looking at it is by differentiating the smoothed data:

$$\frac{\partial}{\partial x} \int_{x-w}^{x+w} G(x-\xi)p(\xi)d\xi = \int_{x-w}^{x+w} G'(x-\xi)p(\xi)d\xi + G(w)(p(x+w) - p(x-w))$$

The last term comes from differentiating the integral's limits and is the same as in the previous expression.

At higher derivatives, more of these term will appear, and our experiments show that the effect of neglecting these terms becomes totally devastating from the second or third derivative up (Table 2). The reason is that truncating the filter amounts to assuming that the data is zero outside the window, and this introduces a sharp artificial discontinuity in the data. The result is a meaninglessly high values for derivatives, especially the high order ones. A small discretization interval aggravates the problem. This "truncation error" is probably a major cause for the failure of the usual methods to obtain reliable derivatives.

Directly calculating these boundary terms is not straightforward. These terms all contain values of the data and/or its derivatives at the ends, and these are not known accurately. For the first derivative one can solve the problem by finding a filter that vanishes at the ends like the one found in the last section, but it will not work at higher orders. In

principle, one can first smooth the image and then perform successive differentiations, but this is quite cumbersome and our experiments with this idea were not very encouraging.

Fortunately, we found a simple “quick fix” to avoid the extra terms above, and it was very successful experimentally in most cases. (In Section 6 we present a more rigorous, but more complicated way to solve the problem.) The “trick” is to smooth the filter so that it, as well as its derivatives, will go down to zero continuously around the ends rather than terminate there discontinuously. In this way all the extra boundary terms above will vanish. The smoothed filter approximates the discontinuous one pretty closely except that its derivatives at the ends can now be handled properly.

This smoothing can be accomplished by a spline polynomial interpolation of the desired order. This calculation does not need to be actually done, it is only a way to understand our method. The only modification we actually do is replacing the derivatives of the truncated filters (which are meaningless at the ends) by central differences. Our experiments show that this is enough to practically eliminate the truncation error. It also provides an easy way around other difficulties such as the discontinuity of our Green function at the center.

We first construct a “piecewise” extended smoothing filter, $G_e(x)$ so that it is identically zero outside a window of width w . Its integral $G_e^{(-1)}$ is thus extended beyond the window as a constant. We can define it on discrete points with interval h as

$$G_e^{(-1)}(i) = \begin{cases} G(ih)^{(-1)} & \text{for } -m \leq i \leq m \\ 1/2 & \text{for } i > m \\ -1/2 & \text{for } i < -m \end{cases} \quad (12)$$

with $m = w/h$ being the farthest point from the window’s center. This filter is equal to $\pm 1/2$ outside the window because of the normalization of G . The difference between these “piecewise” extended filters and the original, either analytic or truncated functions, will be crucial at the ends of the window.

In common smoothing operations, one simply convolves a continuous smoothing filter such as a Gaussian with the data $p(i)$ using the discrete convolution

$$f(i) = \sum_{k=-m}^m hG(kh)p(i-k) = hG \otimes p$$

In effect, the data is regarded as sharp “spikes” at points i with zero everywhere else. We will replace the spikes by square boxes with base h and height proportional to the data value. This will smooth the data considerably, and it can be regarded as a 0-th order spline interpolation. Now we have a continuous (“staircase”) approximation of the data, and one can use the piecewise extended filter $G_e(x)$ to smooth over it. The result can be expressed in terms of the extended filter $G_e^{(-1)}$ as

$$f(i) = \sum_{k=-m}^m (G_e^{(-1)}(k+1/2) - G_e^{(-1)}(k-1/2))p(i-k)$$

We can see that the filter G was replaced by a new smoothing filter, namely the first order central difference of its integral $G_e^{(-1)}$. This can be expressed more concisely as

$$f = DG_e^{(-1)} \otimes p \tag{13}$$

where D denotes the central difference

$$DG(i) = G(i+1/2) - G(i-1/2)$$

This difference can also be expressed as a convolution with the mask \bar{D} :

$$\bar{D} = (-1, 1)$$

So, our new smoothing filter is now $DG_e^{(-1)}$, or $\bar{D} \otimes G_e^{(-1)}$.

For taking derivatives, we need a higher order interpolation. As mentioned before, the higher derivatives are much more sensitive to the discontinuities effects and need smoother interpolation. Thus, for the n -th derivative, we will interpolate the data with an n -th degree B-spline polynomial. These splines have n continuous derivatives, so the error

resulting from the discontinuity problem is for practical purposes eliminated, as confirmed experimentally.

To calculate the spline derivative, we use the well-known result (Ortega and Poole, [1981]) that the n -th order derivative of an n -th order spline interpolation is essentially nothing but the n -th order central difference of the data points. That is,

$$p_n^{(n)}(x) = D^n p(i)/h^n = \bar{D}^n \otimes p(i)/h^n$$

where $p_n(r)$ is the n -th order spline interpolation of $p(i)$ and

$$\bar{D}^n = \bar{D} \otimes \bar{D} \otimes \dots$$

The first few masks \bar{D}^n are easily obtained:

$$\bar{D}^2 = (1, -2, 1)$$

$$\bar{D}^3 = (-1, 3, -3, 1)$$

$$\bar{D}^4 = (1, -4, 6, -4, 1)$$

$$\bar{D}^5 = (-1, 5, -10, 10, -5, 1)$$

Without loss of generality one can set $h = 1$. The transition to $h \neq 1$ can be made by dividing by h^n , since

$$\frac{d^n}{dx^n} = \frac{d^n}{d(ih)^n} = \frac{1}{h^n} \frac{d^n}{di^n}$$

The n -th derivative of the n -th order spline approximation is constant in each interval, yielding another staircase function. We can smooth it as before (eq. 13) by applying the filter $DG_e^{(-1)}$. With the convolutions' associativity and commutativity we have:

$$f^{(n)} = (\bar{D} \otimes G_e^{(-1)}) \otimes (\bar{D}^n \otimes p) = \bar{D}^{n+1} \otimes G_e^{(-1)} \otimes p = D^{n+1} G_e^{(-1)} \otimes p$$

Thus, finally, our n -th order differentiation filter is

$$F^{(n)} = D^{n+1} G_e^{(-1)} \quad (15)$$

or explicitly, with $h \neq 1$,

$$F^n(k) = \bar{D}^{n+1} \otimes G_e^{(-1)} / h^n = \sum_{k=-m-n/2}^{m+n/2} \bar{D}_k^{n+1} G_e^{(-1)}(i-k) / h^n$$

with $n/2$ being rounded to the next lower integer. We see that the usual derivatives of filters have been replaced by central differences of the same order, which is equivalent to a spline interpolation of the filter that smooths its discontinuities up to that order.

We emphasize that it is important to take the central difference of the "piecewise" extended filter $G_e^{(-1)}$ (eq. 12) and not of the differentiable function $G^{(-1)}$ from which it has originated, particularly at the ends. This is what makes the spline approximation of G_e go down to zero at the ends.

It can be easily shown that formally eq. (15) can be generalized to include combinations of differences and derivatives

$$F^{(n)} = D^{n-k} G_e^{(k)}$$

But we have found the above case of $k = -1$ to be the most useful.

6. General Solution for Regularization

In previous sections we found the solution of the regularization problem at the *center* of a window. To find the solution elsewhere we convolved the resulting filter with the image. This has caused problems in the derivatives because it was hard to differentiate the filter at the ends of the window. We used a spline interpolation to smooth this discontinuity. In this section we find an analytic solution for the whole window without any reference to the situation outside. This is a more rigorous solution to the truncation problem than the spline interpolation, and sometimes more accurate, but it is more complicated. It can be useful when the smooth part of the shape is small and there is not much room for a convolution with a wide smoothing filter, or when a large discretization interval is involved making the spline interpolation inaccurate. The boundaries are now the ends of the image, not of a moving window.

The standard methods of mathematical physics usually deal with second order differential equations with only non-mixed boundary conditions (i.e. the conditions at one boundary do not depend on the other). Here we have a fourth order equation (at least) with mixed boundary conditions, so these methods are not much guidance. We found a general technique that solves the problem for any order and boundary conditions (but we do not claim originality).

We start from the general solution of eq. (3):

$$G(x, \xi) = G_{\infty}(x - \xi) + \sum_{i=1}^4 b_i(\xi) H_i(x) \quad (20)$$

The G_{∞} is the solution in the infinite case and it provides the correct jump at $x = \xi$.

$$(1 + \lambda \partial^4) G_{\infty}(x - \xi) = \delta(x - \xi)$$

while the boundary conditions determine the coefficients $b_i(\xi)$ of the homogeneous solutions H_i . These conditions can be expressed by four linear combinations $B_k(G)$, $k = 1 \dots 4$, of all the derivatives $G^{(n)}$ at the boundaries $x = \pm w$:

$$B_k(G) = \sum_{n=1}^4 [\beta_{n,k} G^{(n)}(w, \xi) + \bar{\beta}_{n,k} G^{(n)}(-w, \xi)] = 0$$

where $\beta_{n,k}, \bar{\beta}_{n,k}$ are any given constants. Substituting G from eq. (20) in the above conditions we obtain a system of linear equations for $b_i(\xi)$:

$$\begin{aligned} b_1(\xi) B_1(H_1) + \dots + b_4(\xi) B_1(H_4) &= B_1(G_{\infty}) \\ &\vdots \\ b_1(\xi) B_4(H_1) + \dots + b_4(\xi) B_4(H_4) &= B_4(G_{\infty}) \end{aligned} \quad (21)$$

Since the homogeneous solutions H_i do not depend on ξ , we see that the coefficients multiplying $b_i(\xi)$ on the left hand side are simply constants and the only dependence on ξ

comes from the $B_k(G_\infty)$ on the right hand side. Solving the above system for b_i , we thus obtain a linear combination of the right hand side terms, $B_k(G_\infty)$:

$$b_i(\xi) = \sum_k b_{i,k} B_k(G_\infty(\pm w, \xi))$$

Thus, we have found a Green function which satisfies all conditions. it is easy to see that this is a generalization of the standard Green function method and can be applied to any order and boundary conditions.

Finally, our general solution can be written as

$$f(x) = \int_{-w}^w G(x, \xi) p(\xi) d\xi$$

Since the Green function satisfies the boundary conditions for every ξ , so does the linear combination $f(x)$. For the discrete case, an integrate approximation such as Simpson's rule can be used.

The derivatives at x can be expressed now as

$$f^{(n)} = \int_{-w}^w \frac{\partial}{\partial x} G(x, \xi) p(\xi) d\xi$$

Unlike the filter case, there are no extra terms either from integration by parts or moving boundaries, so the approximation method used before to avoid them is not needed. Attention should be paid, however, to the situation at $x = \xi$, where the third derivative is discontinuous.

Our particular boundary conditions resulted from the preservation of the first four moments (eqs. 7). It is easy to show that these condition can be written linearly as

$$B_1(G) = G'''|_{-w}^w = 0$$

$$B_2(G) = (G' - xG'')|_{-w}^w = 0$$

$$B_3(G) = (G'' - xG''')|_{-w}^w = 0$$

$$B_4(G) = (3G - 3xG' + x^2G''')|_{-w}^w = 0$$

where $f|_{-w}^w$ denotes $f(w) - f(-w)$. (These expressions are simpler when applied to H_i because of [anti]symmetry.) These are the conditions that we substitute in eq. (21) for our particular case.

7. Experiments

We compared the performance of our filters with that of the Gaussian-based ones in various respects. The new filter we used is given in eq. (15), with $G^{(-1)}$ given by eqs. (11,12).

1) Smoothing over a finite interval. Table 1 summarizes the results for various powers x^n , at $x = 1$. The window was wide enough to eliminate truncation errors. For low smoothing, $\sigma = 0.5$, we can see that our filter gives the exact value of 1 up to x^3 , is slightly less accurate for x^4 , and x^5 is reduced about 20%. The Gaussian's errors are growing rapidly. For higher smoothing, $\sigma = 2$, the Gaussian results are totally wrong for powers x^2 and up, in perfect agreement with the theory. Our filter gives correct results up to x^3 . From Section 2, we have that the powers are preserved up to the order l of the filter. The Gaussian's order is 1 and ours here is 3. Section 3 generalizes our method to higher orders.

The small errors that we observe in our method at $\sigma = 2$ have a very systematic behavior. The error in x^2 is independent of x and is about $h^2/12$, while that of x^3 was seen, accordingly, to be $O(h^2x)$. This is exactly the theoretical error for the "mid-point" rule of integration over x^2 and is probably caused by using the analytic boundary conditions of Section 3 rather than a discrete version. In such a version the analytic integrals involved are replaced by sums.

2. Differentiation. Table 2 shows the errors in differentiation produced by truncating the Gaussian at various situations. The derivatives of the Gaussian were used as differentiation filters. We first tested at low smoothing ($\sigma = 0.2$, $h = 0.1$), so that the oversmoothing errors demonstrated above did not play a role. At a window width of 0.6 the Gaussian decayed at the window's end to 0.01 of its peak value, yet the results at high derivatives are totally meaningless. Only when we extended the window such that the decay was 10^{-6} did the results come near the correct values. Our filter decays rather slowly and it would be impractical to achieve this figure. However, taking the central difference instead of the derivative of our filter produce very accurate results on a small window of width 0.5. (This method should solve the truncation problem for the Gaussian too, but than its other errors will show up.)

For stronger smoothing, $\sigma = 2$, $h = .5$, with correspondingly increased window, the Gaussian results were much lower than the actual derivative, because of oversmoothing at the boundary. At a wider window, the errors described in (1) above reappeared and made the results totally off the mark. With our method there is almost no truncation problem and we see that the main characteristics are similar to the previous case: The method gives good results as long as the power of x^n is no more than three above the order of the derivative, consistent with the preservation of the powers up to x^3 . Some truncation error shows up only at high derivatives of high powers, especially at $\frac{d^4}{dx^4}x^7$, but this result is still only 20% off, by far the worst case here. We can see that this error is again proportional to h^2 . For the smaller interval h , the same $\frac{d^4}{dx^4}x^7$ was accurate to 1%. The cause for this inaccuracy is probably the fact that the spline interpolation distorts the high power x^n around the ends more than it does the lower ones. (A method such as described in Section 6 which does not depend on this interpolation should not have this problem.) For functions like $\sin(x/s_0)$ we obtained much better results than the Gaussian with $\sigma \approx s_0$.

3. Random noise. To see the effect of random noise, we can perturb the data by a given amount and observe the effect on the derivatives. We can look at a function such as $f = (x/2)^3$, for which the scale of change as defined in Sec. 2 is 2. We know then that the smoothing parameter σ should not be much greater than 2 to prevent oversmoothing. Thus we look at a filter with $\sigma = 2$, with a window width $w = 5$ and interval $h = 0.5$. If we perturb the data at some pixel i by the amount of Δf , then the error in the derivative at the k -th pixel is $F^{(n)}(i-k)\Delta f$. So, the error is proportional to the magnitude of the filter elements. Table 3 gives the filter elements with the above parameters for the 0, ..., 4-th derivatives. We can see that the values are on average 0.05, giving an error of about 5% in the derivative for an error of $\Delta f = 1$ in the data. The higher derivative filters are no worse than the others in this respect, showing that by choosing the right parameters high derivatives are quite feasible.

The Gaussian derivatives are of the same magnitude, but their systematic errors make them useless at this σ as seen before. The polynomial based filters tend to have higher

values at the center and very low ones towards the ends, making them sensitive to noise at the center and reducing their ability to average out the noise.

8. Conclusions

In this paper we have found the sources of errors in common differentiation filters and have shown ways of overcoming them. The errors we dealt with were (i) Over-smoothing in Gaussian-based filters, leading to inaccurate results even for the simplest functions. We have shown that higher filters can solve the problem. We have proposed a new method of deriving higher order filters that achieves a better balance between smoothing and accuracy than previous ones. (ii) Discontinuities such as truncation of an infinite filter can have devastating effects on the derivatives. We have overcome this by replacing the derivative of the smoothing filter by a central difference of the same order to form the derivative filter. (iii) Random noise can be suppressed without sacrificing accuracy too much, if the parameters of the filter are chosen properly; our "accuracy criterion" gives the relation among the accuracy, the smoothing parameter, and the order of the filter.

Our experiments show that with these techniques one can obtain noise resistant high order derivatives.

References

- Besl, P.J. [1988], "Robust window operators", Proceedings of the Second International Conference on Computer Vision, Tampa, FL.
- Blake, A. and Zisserman, A. [1987], *Visual Reconstruction*, MIT Press, Cambridge, MA.
- Dierckx, P. [1977], "An algorithm for least square fitting of cubic spline surfaces to functions on a rectilinear mesh over a rectangle", *J. Comp. and Appl. Math.*, **3**. 2.
- Grimson, W.E.L. [1981], "The implicit constraint of the primal sketch", MIT A.I. Memo No. 663.

- Haralick, R.M. [1984], "Digital step edges from zero crossings of second directional derivatives", *IEEE Trans. PAMI-6*, 58-68.
- Hashimoto, M. and Sklansky, J. [1987], "Multi-order derivatives for detecting local image characteristics", *Computer Vision, Graphics and Image Processing*, **39**, 28-55.
- Horn, B.K.P. [1983], "The curve of least energy", *ACM Transactions on Mathematical Software*, **9**, 441-460.
- Hueckel, M.F. [1973], "A local visual operator which recognizes edges and lines", *J. Assoc. Comp. Mach.*, **20**, 634-647.
- Hummel, R.A. [1979], "Feature detection using basis functions", *Computer Graphics and Image Processing*, **9**, 40-55.
- Kass, M., Witkin, A., and Terzopoulos, D. [1987], "Snakes, active contour models". *Int. J. Computer Vision*.
- Meer, P. and Weiss, I. [1989], "Smoothed differentiation filters for images". Center for Automation Research, University of Maryland, TR 424.
- Ortega, J.M. and Poole, W.G. [1981], *An Introduction to Numerical Analysis for Differential Equations*, Pitman, Marshfield, MA.
- Poggio, T. [1987], Proceedings of the Image Understanding Workshop, Los Angeles, CA.
- Poggio, T., Torre, T. and Koch, C. [1985], "Computational vision and regularization theory". *Nature*, **317**, 314-319.
- Poggio, T., Voorhees, H. and Yuille, A. [1985], "A regularized solution to edge detection". *J. of Complexity*, **4**, 106-123.
- Weiss, I. [1988], "3-D shape representation by contours". *Computer Vision, Graphics and Image Processing*, **41**, 80-100.
- Weiss, I. [1988], "Projective Invariants of Shapes". Proceedings of the Image Understanding Workshop, Cambridge, MA.

Weiss, I. [1989], "Image Smoothing and Differentiation with Minimal Curvature Filters",
Center for Automation Research, University of Maryland, TR-470.

Weiss, I. [1991], "Noise Resistant Invariants of Curves", *Invariant Workshop*, Iceland.
Center for Automation Research, University of Maryland, TR 537.

Table 1: Smoothing on a Finite Interval

$$s = .5, w = 1, h = 0.1$$

| | 1 | x | x^2 | x^3 | x^4 | x^5 |
|-------|------|------|-------|-------|-------|-------|
| Gauss | 1.00 | 1.00 | 1.25 | 1.75 | 2.68 | 4.44 |
| Ours | 1.00 | 1.00 | 1.00 | 1.00 | 0.96 | 0.80 |

$$s = 2, w = 5, h = 0.5$$

| | 1 | x | x^2 | x^3 | x^4 |
|-------|------|------|-------|-------|--------|
| Gauss | 1.00 | 1.00 | 4.99 | 12.99 | 72.99 |
| Ours | 1.00 | 1.00 | 1.02 | 1.06 | -23.27 |

Table 2: Truncation Error

$$s = .2, h = .1$$

| | w | $d^0 x^2$ | $d^0 x^4$ | $d^4 x^4$ | $d^4 x^5$ | $d^4 x^6$ | $d^4 x^7$ |
|---------|-----|-----------|-----------|-----------|-----------|-----------|-----------|
| Trunc. | 0.6 | 1.03 | 1.24 | -252 | -361 | -466 | 530 |
| Gauss | 1.0 | 1.04 | 1.25 | 23.4 | 118.9 | 372.0 | 935 |
| Ours | .5 | 1.0 | 1.0 | 24.0 | 120.0 | 361 | 850 |
| correct | | 1 | 1 | 24 | 120 | 360 | 840 |

$$s = 2, h = .5$$

| | w | $d^0 x^2$ | $d^0 x^4$ | $d^4 x^4$ | $d^4 x^5$ | $d^4 x^6$ | $d^4 x^7$ |
|--------|-----|-----------|-----------|-----------|-----------|-----------|-----------|
| Trunc. | 0.6 | 4.9 | 68.6 | 0.19 | 9.04 | 281 | 1821 |
| Gauss | 1.0 | 1.99 | 72.9 | 23.8 | 119.2 | 1780.4 | 10793 |
| Ours | .5 | 1.02 | -23.27 | 24.0 | 120.0 | 397.4 | 1101.6 |

Table 3: The Differentiation Filters

 $s : 2, w = 5, h = .5$

| d^0 | d^1 | d^2 | d^3 | d^4 |
|---------------|---------------|---------------|-------------|---------------|
| 0.15938313 | 0.0 | -0.07757177 | 0.0 | 0.1673013 |
| 0.14968666 | -0.035587692 | -0.056659108 | 0.059712945 | 0.031345331 |
| 0.12582541 | -0.056247147 | -0.027910113 | 0.049725615 | -0.030269221 |
| 0.094986638 | -0.064475199 | -0.0067284237 | 0.035947677 | -0.023677088 |
| 0.062465755 | -0.063716331 | 0.0085339938 | 0.026105913 | -0.015705673 |
| 0.032078371 | -0.056430985 | 0.019869993 | 0.020178524 | -0.0082267441 |
| 0.0066584857 | -0.044101009 | 0.029149307 | 0.017725599 | -0.0019682933 |
| -0.011474073 | -0.027339632 | 0.037936547 | 0.017982888 | 0.0024734119 |
| -0.020122496 | -0.0060825336 | 0.04734214 | 0.01990211 | -0.052793667 |
| -0.016935383 | 0.020150092 | 0.043549316 | -0.10480257 | -0.33844196 |
| -0.0028609419 | 0.020182075 | -0.044853998 | -0.08085623 | 0.48725418 |
| - | - | -0.011443767 | 0.0807283 | -0.087865854 |
| - | - | - | - | -0.04577507 |

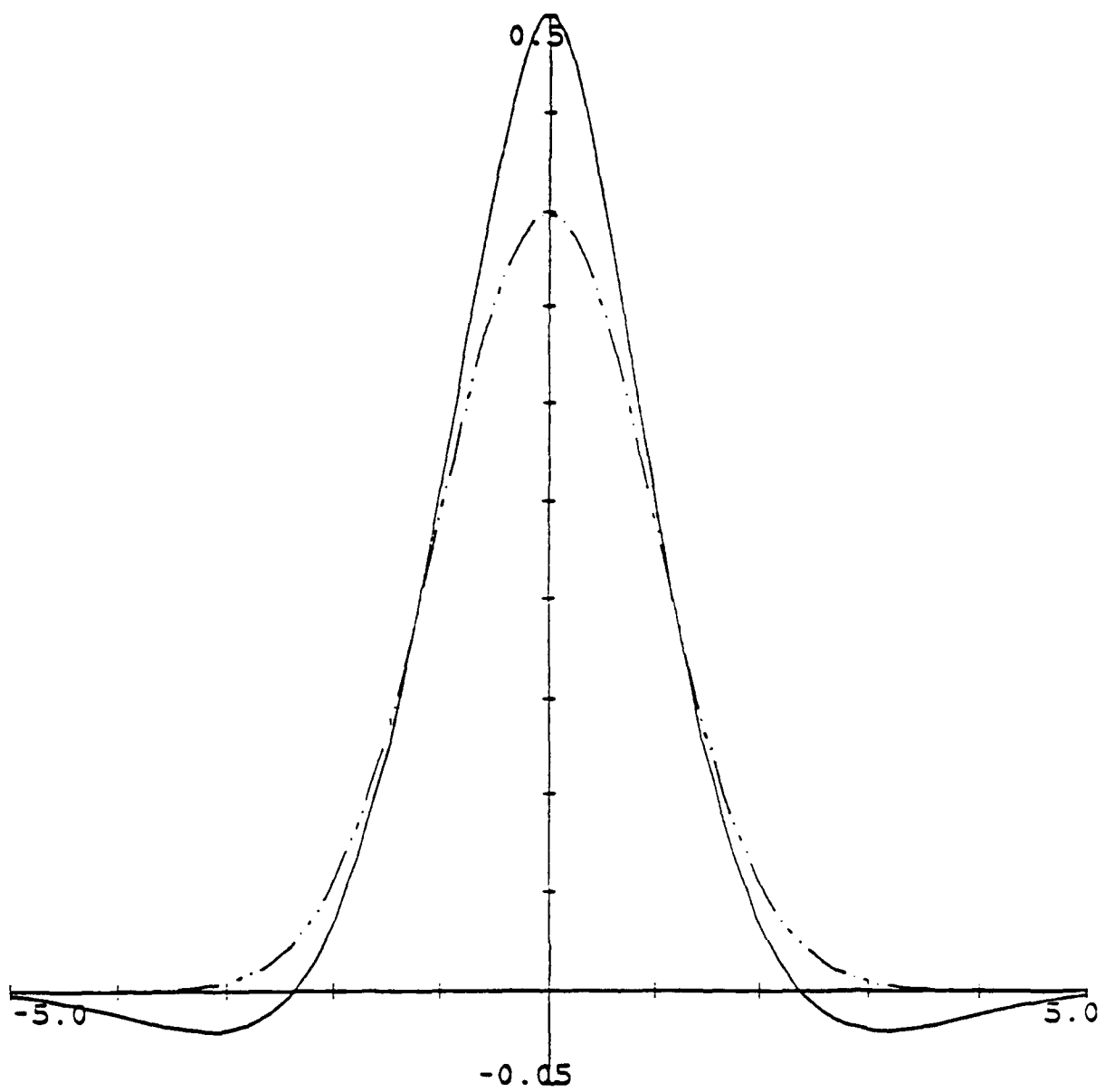


Figure 1: Smoothing Filters: Minimal-Curvature and Gaussian

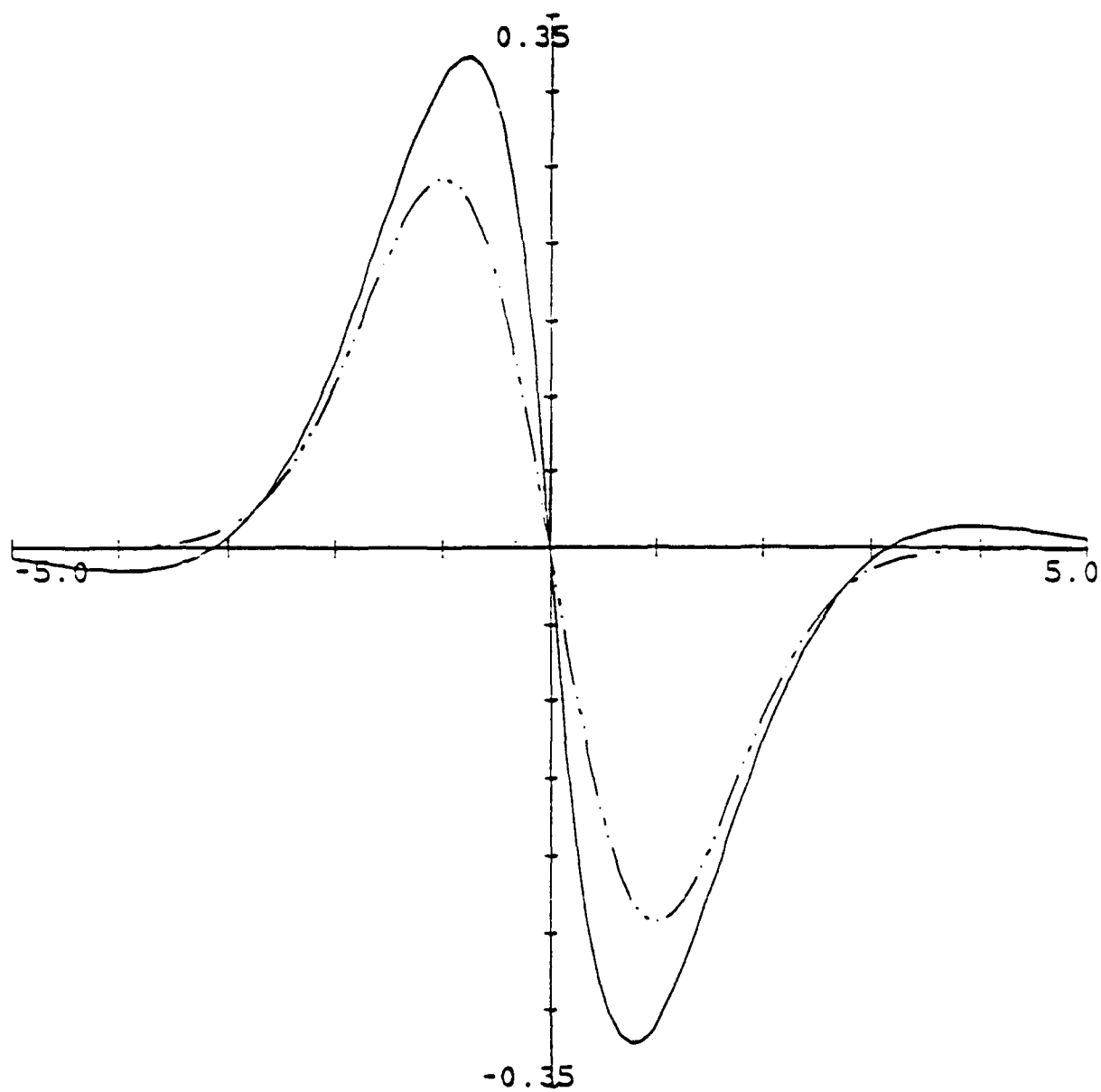


Figure 2: Differentiation Filters: Minimal-Curvature and Gaussian

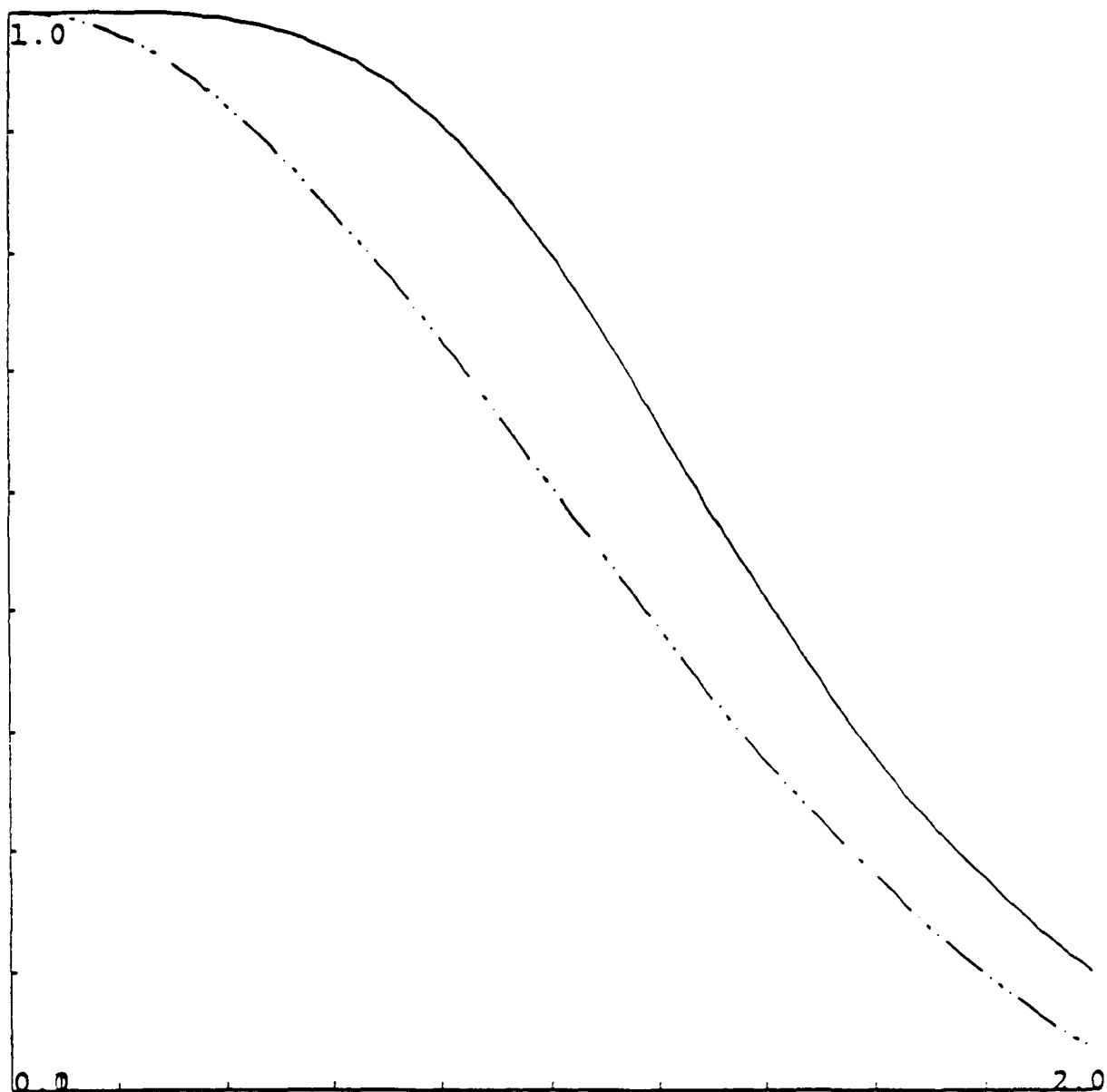


Figure 3: Fourier Transform of Smoothing Filters

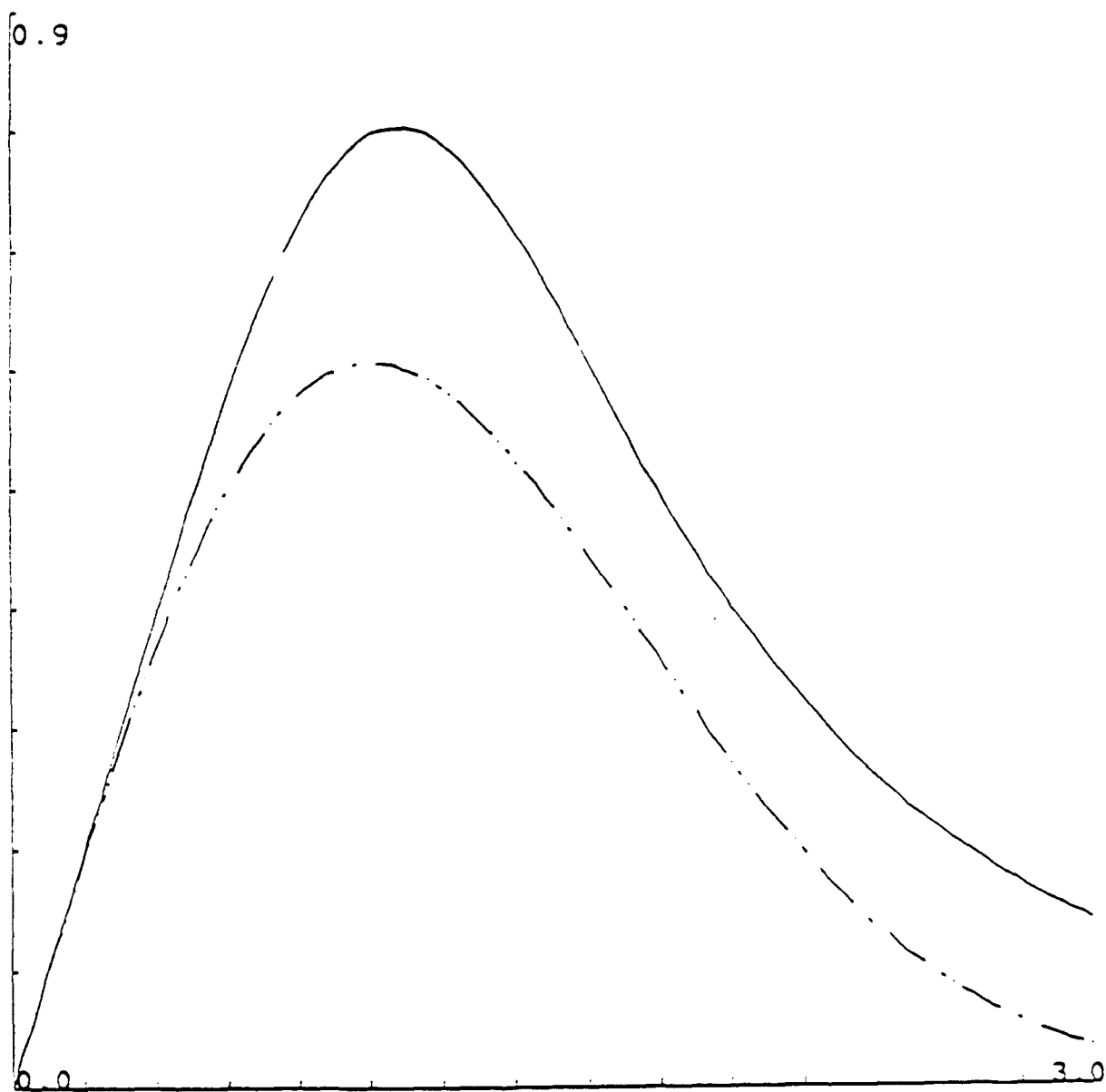


Figure 4: Fourier Transform of Differentiation Filters

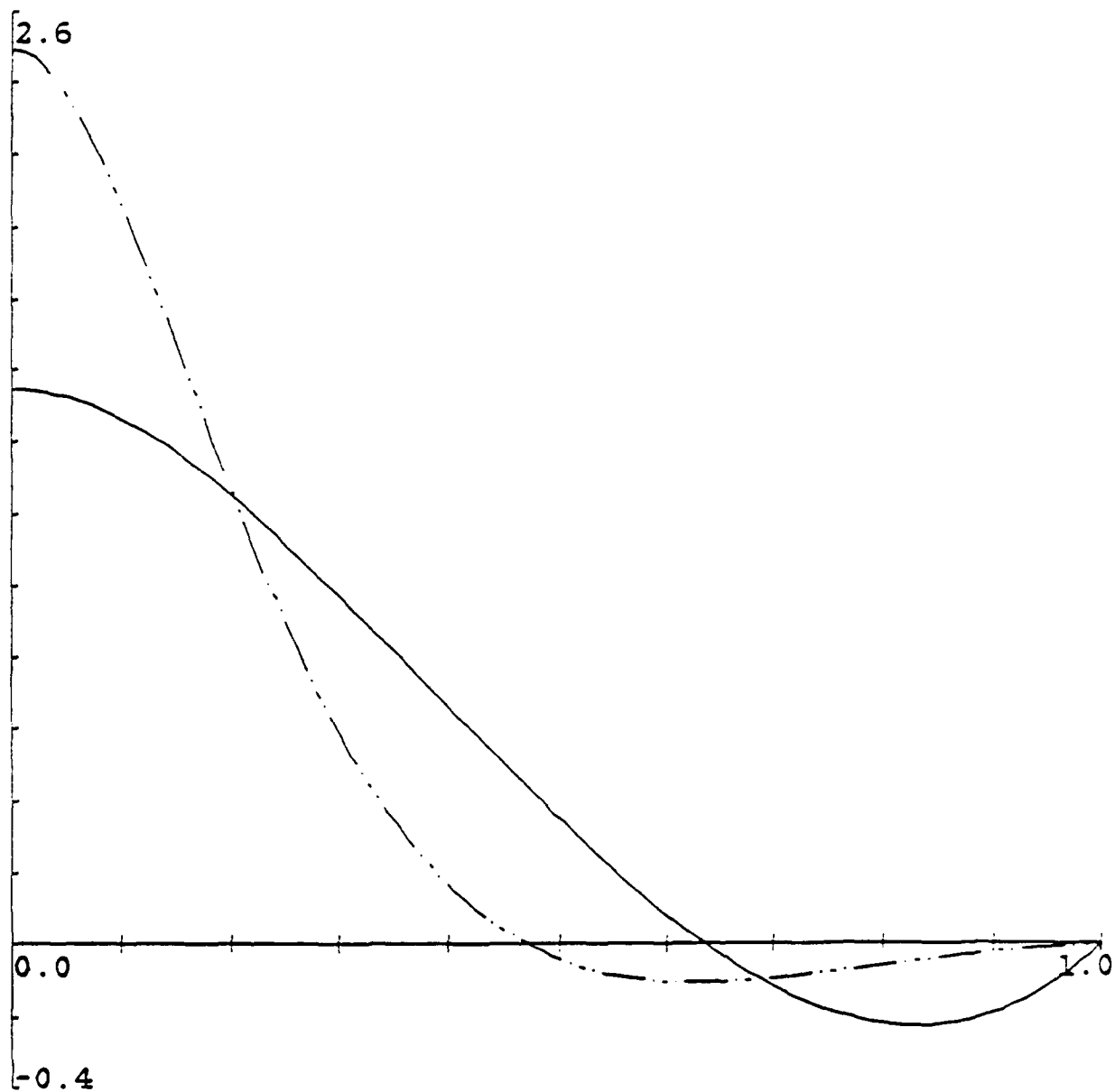


Figure 5: Finite Smoothing Filters

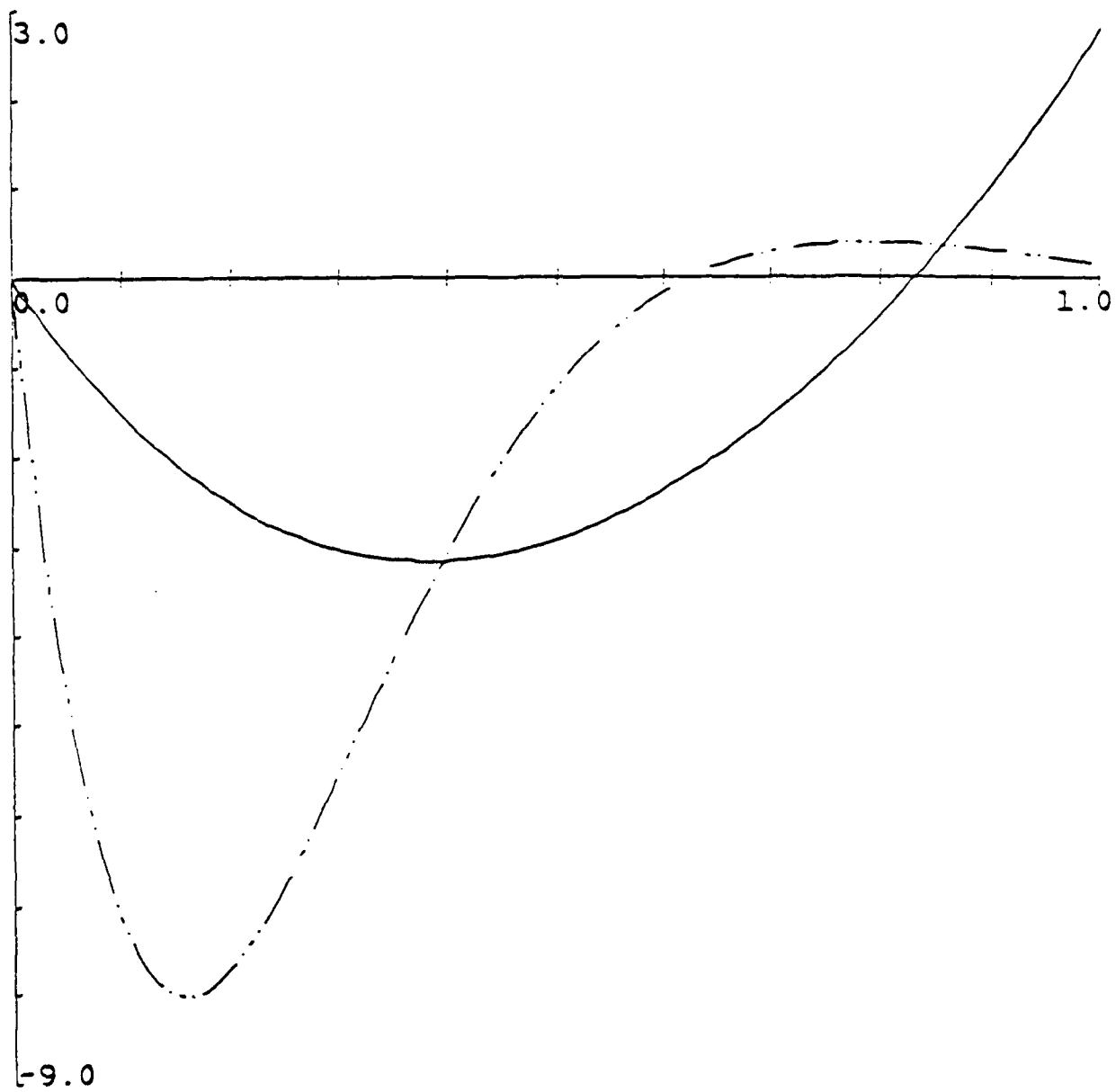


Figure 6: Finite Differentiation Filters

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| | | | | |
|---|---|--|---|--|
| 1. AGENCY USE ONLY (Leave blank) | | 2. REPORT DATE March 1991 | 3. REPORT TYPE AND DATES COVERED Technical | |
| 4. TITLE AND SUBTITLE HIGH ORDER DIFFERENTIATION FILTERS THAT WORK | | | 5. FUNDING NUMBERS N00014-91-J-1222 | |
| 6. AUTHOR(S) Isaac Weiss | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Center for Automation Research University of Maryland College Park, MD 20742-3411 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER CAR-TR-545 CS-TR-2639 | |
| 9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Department of the Navy Office of the Chief of Naval Research 800 North Quincy Street, Code 1513:BSM Arlington, VA 22217-5000 | | | 10. SPONSORING / MONITORING AGENCY REPORT NUMBER | |
| 11. SUPPLEMENTARY NOTES | | | | |
| 12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited | | | 12b. DISTRIBUTION CODE | |
| 13. ABSTRACT (Maximum 200 words) Reliable derivatives of digital images have always been hard to obtain, especially (but not only) at high orders. We present new filters that give more accurate derivatives than the traditional Gaussian ones. We show that the traditional filters give incorrect derivatives even for an analytic, noiseless, infinite image, because they smooth the image too much. For a finite interval, the effects of truncating the filter become intolerable for high derivatives. We derive filters that allow a higher amount of noise suppression with less compromise of accuracy than the Gaussian. The filters are easy to compute at arbitrary size. In addition, a general analytic (non-filter) solution is derived for the regularization problem on a finite interval. | | | | |
| 14. SUBJECT TERMS smoothing, differentiation, regularizaiton, filters, fitting | | | 15. NUMBER OF PAGES 38 | |
| | | | 16. PRICE CODE | |
| 17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED | 18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED | 19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED | 20. LIMITATION OF ABSTRACT SAR | |